Atty. Docket No.: 30390-14

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Ken UMENO	Group Art Unit:	Not Assigned
Serial No.:	10/803,587	Examiner:	Not Assigned
Filed:	17 March 2004		
For:	RANDOM SEQUENCE GENERATING APPARATUS, ENCRYPTION/DECRYPTION APPARATUS, RANDOM SEQUENCE GENERATING METHOD, ENCRYPTION/ DECRYPTION METHOD AND PROGRAM		

San Diego, California
March 26, 2004

Mail Stop: PATENT APPLICATION
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Dear Sir or Madam:

Transmitted herewith is Priority Document: Japanese Application Serial No. JP 2003-075438. Although it is believed that no fees are due for this submission, the Commissioner is authorized to charge any shortage in fees due in connection with the filing of this paper to our Deposit Account No. 50-2298 in the name of Luce, Forward, Hamilton & Scripps LLP.

Respectfully submitted,

3/26/04
Date

Mitchell P. Brook
Attorney for Applicant(s)
Reg. No. 32,967
c/o LUCE, FORWARD, HAMILTON & SCRIPPS LLP
11988 El Camino Real, Suite 200
San Diego, California 92130
Telephone No.: (858) 720-6300

CERTIFICATE OF MAILING

I hereby certify that this correspondence, and anything referred to as transmitted herewith, is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

Date: 26 Mar. 2004

By: Amy M. Sheridan

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　３月１９日

出願番号
Application Number: 特願２００３－０７５４３８

[ST. 10/C]: [ＪＰ２００３－０７５４３８]

出願人
Applicant(s): 独立行政法人通信総合研究所

２００４年　２月１７日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号　出証特２００４－３０１０２２９

【書類名】　特許願
【整理番号】　CRL-02-43
【提出日】　平成15年 3月19日
【あて先】　特許庁長官　殿
【国際特許分類】　H04B　1/00
【発明者】
　【住所又は居所】　東京都小金井市貫井北町４－２－１　独立行政法人通信総合研究所内
　【氏名】　梅野　健
【特許出願人】
　【識別番号】　301022471
　【氏名又は名称】　独立行政法人通信総合研究所
【代理人】
　【識別番号】　100095407
　【弁理士】
　【氏名又は名称】　木村　満
【選任した代理人】
　【識別番号】　100110135
　【弁理士】
　【氏名又は名称】　石井　裕一郎
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　0112098
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　乱数列生成装置、暗号化復号化装置、乱数列生成方法、暗号化復号化方法、ならびに、プログラム

【特許請求の範囲】

【請求項１】

wビットの乱数の列を生成する乱数列生成装置であって、種受付部と、初期化部と、変換部と、回転部と、更新部と、出力部と、を備え、整数n，m（$1 \leqq n \leqq m-1$）に対して、

前記種受付部は、wビットの整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を種として受け付け、

前記初期化部は、当該受け付けられた整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を、整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として前記変換部に与え、

前記変換部は、当該与えられた整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$のそれぞれに対して所定の変換を施してwビットの整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$を得て、

前記回転部は、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$をwmビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を得て、

前記更新部は、当該整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として前記変換部に与え、

前記出力部は、前記変換部における変換ならびに前記回転部における回転が、所定の回数繰り返された場合、最後に得られた$z_1, z_2, \cdots, z_n$を、乱数列$r_1, r_2, \cdots, r_n$として出力する

ことを特徴とするもの。

【請求項２】

請求項１に記載の乱数列生成装置であって、

前記変換部は、写像$g(\cdot, \cdot)$を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とするもの。

【請求項３】

請求項１に記載の乱数列生成装置であって、

前記変換部は、所定の整数cと、写像g(・,・)と、を用いて、整数i（1≦i≦m-1)についての漸化式

$y_1 = g(c, x_1)$；

$y_{i+1} = g(y_i, x_{i+1})$

により変換を行う

ことを特徴とするもの。

【請求項４】

請求項１に記載の乱数列生成装置であって、

前記変換部は、写像g(・,・)を用いて、整数i（1≦i≦m-1)についての漸化式

$y_1 = g(c, x_1)$；

$y_{i+1} = g(x_i, x_{i+1})$

により変換を行う

ことを特徴とするもの。

【請求項５】

請求項２から４のいずれか１項に記載の乱数列生成装置であって、

当該写像g(・,・)は、所定の写像h(・)と所定の整数q（0≦q≦2^w-1)により

$g(a,b) = 2b^2 + h(a)b + q \pmod{2^w}$

と定義される

ことを特徴とするもの。

【請求項６】

請求項５に記載の乱数列生成装置であって、

当該写像h(・)は、

$h(a) = a$

と定義される

ことを特徴とするもの。

【請求項７】

請求項５に記載の乱数列生成装置であって、

当該写像h(・)は、与えられた値の数値表現の所定のビットをクリアする演算により定義される

ことを特徴とするもの。

【請求項８】

請求項５に記載の乱数列生成装置であって、

当該写像h(・)は、与えられた値の数値表現の所定のビットを反転する演算により定義される

ことを特徴とするもの。

【請求項９】

請求項５に記載の乱数列生成装置であって、

当該写像h(・)は、与えられた値の数値表現の最下位２ビットを０１に置換する演算により定義される

ことを特徴とするもの。

【請求項１０】

請求項１から９のいずれか１項に記載の乱数列生成装置であって、

前記回転部は、当該整数の列$y_{n+1}, \cdots, y_m$を$w(m-n+1)$ビットのビット列と見て、これから所定の位置のビットを１個以上抽出して並べたビット列を整数と見たときの整数値を、回転ビット数として得る

ことを特徴とするもの。

【請求項１１】

請求項１０に記載の乱数列生成装置であって、

前記回転部は、当該整数の列$y_{n+1}, \cdots, y_m$を$w(m-n+1)$ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決める

ことを特徴とするもの。

【請求項１２】

請求項１から１１のいずれか１項に記載の乱数生成装置であって、

前記回転部は、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n$をwnビットのビット列と見て、これに対して当該得られた回転ビ

ット数の回転演算を行って、得られたwnビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n$を得て、整数の列$y_{n+1}, \cdots, y_m$をw(m-n)ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られたw(m-n)ビットのビット列からwビットの整数の列$z_{n+1}, \cdots, z_m$を得る

ことを特徴とするもの。

【請求項１３】

乱数生成部と、メッセージ受付部と、暗号化復号化部と、を備える暗号化復号化装置であって、

前記乱数生成部は、請求項１から１２のいずれか１項に記載の乱数生成装置により、乱数列$r_1, r_2, \cdots, r_n$を生成し、

前記メッセージ受付部は、wビットの整数の整数列$p_1, p_2, \cdots,$をメッセージとして受け付け、

前記暗号化復号化部は、wビットの整数の列p_1 xor r_1, p_2 xor $r_2, \cdots, p_i$ xor r((i+n-1) mod n) + 1,…を暗号化もしくは復号化の結果として出力する

ことを特徴とするもの。

【請求項１４】

wビットの乱数の列を生成する乱数列生成方法であって、種受付工程と、初期化工程と、変換工程と、回転工程と、更新工程と、出力工程と、を備え、整数n, m (1≦n≦m-1)に対して、

前記種受付工程では、wビットの整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を種として受け付け、

前記初期化工程では、当該受け付けられた整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を、整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として前記変換工程に与え、

前記変換工程では、当該与えられた整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$のそれぞれに対して所定の変換を施してwビットの整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$を得て、

前記回転工程では、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$をwmビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を得て、

前記更新工程では、当該整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として前記変換工程に与え、

前記出力工程では、前記変換工程における変換ならびに前記回転工程における回転が、所定の回数繰り返された場合、最後に得られた$z_1, z_2, \cdots, z_n$を、乱数列$r_1, r_2, \cdots, r_n$として出力する

ことを特徴とする方法。

【請求項１５】

請求項１４に記載の乱数列生成方法であって、

前記変換工程では、写像$g(\cdot, \cdot)$を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とする方法。

【請求項１６】

請求項１４に記載の乱数列生成方法であって、

前記変換工程では、所定の整数cと、写像$g(\cdot, \cdot)$と、を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行う

ことを特徴とする方法。

【請求項１７】

請求項１４に記載の乱数列生成方法であって、

前記変換工程では、写像$g(\cdot, \cdot)$を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とする方法。

【請求項１８】

請求項１５から１７のいずれか１項に記載の乱数列生成方法であって、

当該写像g(・,・)は、所定の写像h(・)と所定の整数q (0≦q≦2^w-1)により

$$g(a,b) = 2b^2 + h(a)b + q \pmod{2^w}$$

と定義される

ことを特徴とする方法。

【請求項１９】

請求項１８に記載の乱数列生成方法であって、

当該写像h(・)は、

$$h(a) = a$$

と定義される

ことを特徴とする方法。

【請求項２０】

請求項１８に記載の乱数列生成方法であって、

当該写像h(・)は、与えられた値の数値表現の所定のビットをクリアする演算により定義される

ことを特徴とする方法。

【請求項２１】

請求項１８に記載の乱数列生成方法であって、

当該写像h(・)は、与えられた値の数値表現の所定のビットを反転する演算により定義される

ことを特徴とする方法。

【請求項２２】

請求項１８に記載の乱数列生成方法であって、

当該写像h(・)は、与えられた値の数値表現の最下位２ビットを０１に置換する演算により定義される

ことを特徴とする方法。

【請求項２３】

請求項１４から２２のいずれか１項に記載の乱数列生成方法であって、

前記回転工程では、当該整数の列$y_{n+1}, \cdots, y_m$をw(m-n+1)ビットのビット列と見て、これから所定の位置のビットを１個以上抽出して並べたビット列を整数と見たときの整数値を、回転ビット数として得る

ことを特徴とする方法。

【請求項２４】

請求項２３に記載の乱数列生成方法であって、

前記回転工程では、当該整数の列$y_{n+1}, \cdots, y_m$をw(m-n+1)ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決める

ことを特徴とする方法。

【請求項２５】

請求項１４から２４のいずれか１項に記載の乱数生成方法であって、

前記回転工程では、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n$をwnビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られたwnビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n$を得て、整数の列$y_{n+1}, \cdots, y_m$をw(m-n)ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られたw(m-n)ビットのビット列からwビットの整数の列$z_{n+1}, \cdots, z_m$を得る

ことを特徴とする方法。

【請求項２６】

乱数生成工程と、メッセージ受付工程と、暗号化復号化工程と、を備える暗号化復号化方法であって、

前記乱数生成工程では、請求項１３から２３のいずれか１項に記載の乱数生成方法により、乱数列$r_1, r_2, \cdots, r_n$を生成し、

前記メッセージ受付工程では、wビットの整数の整数列$p_1, p_2, \cdots,$をメッセージとして受け付け、

前記暗号化復号化工程では、wビットの整数の列p_1 xor r_1, p_2 xor $r_2, \cdots, p_i$ xor $r((i+n-1) \bmod n) + 1, \cdots$を暗号化もしくは復号化の結果として出力する

ことを特徴とする方法。

【請求項２７】

コンピュータを、請求項１から１２のいずれか１項に記載の乱数列生成装置として機能させることを特徴とするプログラム。

【請求項２８】

コンピュータを、請求項１３に記載の暗号化復号化装置として機能させることを特徴とするプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、乱数列生成装置、暗号化復号化装置、乱数列生成方法、暗号化復号化方法、ならびに、プログラムに関する。

【０００２】

【従来の技術】

従来から、種々の乱数列生成のための技術が提案されている。これらの技術によって得られた乱数は、たとえば、モンテカルロ法による各種の物理現象、化学現象などの模擬実験や、秘密通信のブロック暗号システムにおいて用いられる。

【０００３】

【発明が解決しようとする課題】

さて、このような乱数列の生成技術においては、得られた乱数列に含まれる数値の分布ができるだけ一様であることや、当該数値のコンピュータにおける数値表現の所定のビットのみを見た場合に、当該ビットの「０」と「１」の出現頻度にできるだけ偏りがないことや、乱数列の周期ができるだけ長いことなど、種々の性質を満たすことが望ましい。

【０００４】

本発明は、生成される乱数列が乱数列として好ましい性質を有するような乱数列生成装置、乱数列生成方法、ならびに、これらを用いた暗号化復号化装置、暗号化復号化方法、ならびに、これらをコンピュータによって実現するためのプログラムを提供することを目的とする。

【０００５】

【課題を解決するための手段】

以上の目的を達成するため、本発明の原理にしたがって、下記の発明を開示する。

【０００６】

本発明の第１の観点に係る乱数列生成装置は、wビットの乱数の列を生成し、種受付部と、初期化部と、変換部と、回転部と、更新部と、出力部と、を備え、整数n，m（$1 \leqq n \leqq m-1$）に対して、以下のように構成する。

【０００７】

すなわち、種受付部は、wビットの整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を種として受け付ける。

【０００８】

一方、初期化部は、当該受け付けられた整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を、整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として変換部に与える。

【０００９】

さらに、変換部は、当該与えられた整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$のそれぞれに対して所定の変換を施してwビットの整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$を得る。

【００１０】

そして、回転部は、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$をwmビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を得る。

【００１１】

一方、更新部は、当該整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として変換部に与える。

【００１２】

さらに、出力部は、変換部における変換ならびに回転部における回転が、所定の回数繰り返された場合、最後に得られた$z_1, z_2, \cdots, z_n$を、乱数列$r_1, r_2, \cdots, r_n$として出力する。

【００１３】

また、本発明の乱数列生成装置において、変換部は、写像g(・,・)を用いて、整数i（1≦i≦m-1)についての漸化式

$y_1 = g(x_m, x_1)$；

$y_{i+1} = g(x_i, x_{i+1})$

により変換を行うように構成することができる。

【００１４】

また、本発明の乱数列生成装置において、変換部は、所定の整数cと、写像g(・,・)と、を用いて、整数i（1≦i≦m-1)についての漸化式

$y_1 = g(c, x_1)$；

$y_{i+1} = g(y_i, x_{i+1})$

により変換を行うように構成することができる。

【００１５】

また、本発明の乱数列生成装置において、変換部は、写像g(・,・)を用いて、整数i（1≦i≦m-1)についての漸化式

$y_1 = g(c, x_1)$；

$y_{i+1} = g(x_i, x_{i+1})$

により変換を行うように構成することができる。

【００１６】

また、本発明の乱数列生成装置において、当該写像g(・,・)は、所定の写像h(・)と所定の整数q（0≦q≦2^w-1)により

$g(a,b) = 2b^2 + h(a)b + q \pmod{2^w}$

と定義されるように構成することができる。

【００１７】

また、本発明の乱数列生成装置において、当該写像h(・)は、

$h(a) = a$

と定義されるように構成することができる。

【００１８】

また、本発明の乱数列生成装置において、当該写像h(・)は、与えられた値の数値表現の所定のビットをクリアする演算により定義されるように構成すること

ができる。

【００１９】

また、本発明の乱数列生成装置において、当該写像h(・)は、与えられた値の数値表現の所定のビットを反転する演算により定義されるように構成することができる。

【００２０】

また、本発明の乱数列生成装置において、当該写像h(・)は、与えられた値の数値表現の最下位２ビットを０１に置換する演算により定義されるように構成することができる。

【００２１】

また、本発明の乱数列生成装置において、回転部は、当該整数の列$y_{n+1}, \cdots, y_m$を$w(m-n+1)$ビットのビット列と見て、これから所定の位置のビットを１個以上抽出して並べたビット列を整数と見たときの整数値を、回転ビット数として得るように構成することができる。

【００２２】

また、本発明の乱数列生成装置において、回転部は、当該整数の列$y_{n+1}, \cdots, y_m$を$w(m-n+1)$ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決めるように構成することができる。

【００２３】

また、本発明の乱数列生成装置において、回転部は、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n$をwnビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られたwnビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n$を得て、整数の列$y_{n+1}, \cdots, y_m$を$w(m-n)$ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた$w(m-n)$ビットのビット列からwビットの整数の列$z_{n+1}, \cdots, z_m$を得るように構成することができる。すなわち、z_iは、u_iを所定の回転ビット数だけ回転演算したものである。

【００２４】

本発明の他の観点に係る暗号化復号化装置は、乱数生成部と、メッセージ受付

部と、暗号化復号化部と、を備え、以下のように構成する。

【００２５】

すなわち、乱数生成部は、上記の乱数生成装置により、乱数列$r_1, r_2, \cdots, r_n$を生成する。

【００２６】

一方、メッセージ受付部は、wビットの整数の整数列$p_1, p_2, \cdots,$をメッセージとして受け付ける。

【００２７】

さらに、暗号化復号化部は、wビットの整数の列p_1 xor r_1, p_2 xor $r_2, \cdots, p_i$ xor $r((i+n-1) \bmod n) + 1, \cdots$を暗号化もしくは復号化の結果として出力する。

【００２８】

本発明の他の観点に係る乱数列生成方法は、wビットの乱数の列を生成し、種受付工程と、初期化工程と、変換工程と、回転工程と、更新工程と、出力工程と、を備え、整数n, m $(1 \leqq n \leqq m-1)$に対して、以下のように構成する。

【００２９】

すなわち、種受付工程では、wビットの整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を種として受け付ける。

【００３０】

一方、初期化工程では、当該受け付けられた整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を、整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として変換工程に与える。

【００３１】

さらに、変換工程では、当該与えられた整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$のそれぞれに対して所定の変換を施してwビットの整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$を得る。

【００３２】

そして、回転工程では、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$をwmビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を得る。

【００３３】

一方、更新工程では、当該整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として変換工程に与える。

【００３４】

さらに、出力工程では、変換工程における変換ならびに回転工程における回転が、所定の回数繰り返された場合、最後に得られた$z_1, z_2, \cdots, z_n$を、乱数列$r_1, r_2, \cdots, r_n$として出力する。

【００３５】

また、本発明の乱数列生成方法において、変換工程では、写像$g(\cdot, \cdot)$を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行うように構成することができる。

【００３６】

また、本発明の乱数列生成方法において、変換工程では、所定の整数cと、写像$g(\cdot, \cdot)$と、を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行うように構成することができる。

【００３７】

また、本発明の乱数列生成方法において、変換工程では、写像$g(\cdot, \cdot)$を用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行うように構成することができる。

【００３８】

また、本発明の乱数列生成方法において、当該写像$g(\cdot, \cdot)$は、所定の写像$h(\cdot)$と所定の整数q（$0 \leqq q \leqq 2^w-1$）により

$$g(a,b) = 2b^2 + h(a)b + q \pmod{2^w}$$

と定義されるように構成することができる。

【００３９】

また、本発明の乱数列生成方法において、当該写像h(・)は、

$h(a) = a$

と定義されるように構成することができる。

【００４０】

また、本発明の乱数列生成方法において、当該写像h(・)は、与えられた値の数値表現の所定のビットをクリアする演算により定義されるように構成することができる。

【００４１】

また、本発明の乱数列生成方法において、当該写像h(・)は、与えられた値の数値表現の所定のビットを反転する演算により定義されるように構成することができる。

【００４２】

また、本発明の乱数列生成方法において、当該写像h(・)は、与えられた値の数値表現の最下位２ビットを０１に置換する演算により定義されるように構成することができる。

【００４３】

また、本発明の乱数列生成方法において、回転工程では、当該整数の列y_{n+1}, …, y_mを$w(m-n+1)$ビットのビット列と見て、これから所定の位置のビットを１個以上抽出して並べたビット列を整数と見たときの整数値を、回転ビット数として得るように構成することができる。

【００４４】

また、本発明の乱数列生成方法において、回転工程では、当該整数の列y_{n+1}, …, y_mを$w(m-n+1)$ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決めるように構成することができる。

【００４５】

また、本発明の乱数列生成方法において、回転工程では、当該整数の列y_{n+1}, …, y_mから回転ビット数を得て、当該整数の列$y_1, y_2, \cdots, y_n$をw_nビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた

wnビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n$を得て、整数の列y_{n+1}, $\cdots, y_m$をw(m-n)ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られたw(m-n)ビットのビット列からwビットの整数の列$z_{n+1}, \cdots, z_m$を得るように構成することができる。すなわち、z_iは、u_iを所定の回転ビット数だけ回転演算したものである。

【００４６】

本発明の他の観点に係る暗号化復号化方法は、乱数生成工程と、メッセージ受付工程と、暗号化復号化工程と、を備え、以下のように構成する。

【００４７】

すなわち、乱数生成工程では、上記の乱数生成方法により、乱数列$r_1, r_2, \cdots, r_n$を生成する。

【００４８】

一方、メッセージ受付工程では、wビットの整数の整数列$p_1, p_2, \cdots,$をメッセージとして受け付ける。

【００４９】

さらに、暗号化復号化工程では、wビットの整数の列p_1 xor r_1, p_2 xor $r_2, \cdots,$ p_i xor $r((i+n-1) \bmod n) + 1, \cdots$を暗号化もしくは復号化の結果として出力する。

【００５０】

本発明の他の観点に係るプログラムは、コンピュータを、上記の乱数列生成装置もしくは暗号化復号化装置として機能させ、もしくは、コンピュータに、上記の乱数列生成方法もしくは暗号化復号化方法を実行させるように構成する。

【００５１】

これらのプログラムは、コンパクトディスク、フレキシブルディスク、ハードディスク、光磁気ディスク、ディジタルビデオディスク、磁気テープ、半導体メモリ等のコンピュータ読取可能な情報記録媒体に記録することができる。

【００５２】

上記プログラムは、当該プログラムが実行されるコンピュータとは独立して、コンピュータ通信網を介して配布・販売することができる。また、上記情報記録

媒体は、当該コンピュータとは独立して配布・販売することができる。

【００５３】

【発明の実施の形態】

以下に本発明の実施形態を説明する。なお、以下に説明する実施形態は説明のためのものであり、本願発明の範囲を制限するものではない。したがって、当業者であればこれらの各要素もしくは全要素をこれと均等なものに置換した実施形態を採用することが可能であるが、これらの実施形態も本願発明の範囲に含まれる。

【００５４】

（発明の実施の形態）

以下で説明する本発明の実施形態においては、「wビットの数値表現による乱数」の列を生成するために、有限体上の非線型変換として、所定の写像h(・)と所定の整数q（$0 \leqq q \leqq 2^w-1$）とを用いて

$$g(a,b) = 2b^2 + h(a)b + q \pmod{2^w}$$

により定義される写像g(・,・)を用いる。

【００５５】

写像h(・)としては、恒等写像

$$h(a) = a$$

を用いることができる。

【００５６】

また、所定のマスク値MASKを利用することにより、与えられた値aの数値表現の所定のビットをクリアする演算

$$h(a) = a \text{ and } MASK$$

や、所定のビットを反転する演算

$$h(a) = a \text{ xor } MASK$$

を採用してもよい。

【００５７】

さらに、最下位の２ビットを値０１に変換する演算

$$h(a) = (a \text{ and } (\text{not } 3)) \text{ or } 1$$

などを採用することができる。

【００５８】

ここで、各演算子は値aの数値表現（整数表現）に対するもので、andはビット積（ビットアンド）、xorはビット排他的和（ビットエクスクルーシブオア）、notはビット反転（ビットノット）、orはビット和（ビットオア）にそれぞれ相当するものである。

【００５９】

したがって、これらの演算は、コンピュータにおいては桁上がりや桁下がりなどをとりたてて考慮せずに、wビットの整数演算として用意されているものをそのまま適用して実現することができる。

【００６０】

またwの値は、当該コンピュータのＣＰＵ（Central Processing Unit）に用意されている機械語のビット幅、もしくは、これよりも小さい幅に対応させることが望ましい。

【００６１】

現在のところ、世界最高速のブロック暗号技術といわれているＲＣ６は、有限体上の非線型変換

$$f(x) = 2x^2 + x \pmod{2^w}$$

を用いることによって実現されており、これによって、ある種から生成される乱数列は、これとは異なる種から生成される乱数列とは常に異なるものであり（１対１性）、生成される乱数列の最長周期が2^{w-1}となっている。

【００６２】

本実施形態において採用される写像g(・,・)は、ＲＣ６における有限体上の非線型変換をさらに一般化したものであり、

$$h(a) = 1;$$

$$q = 0$$

としたg(・,・)を採用した場合には、ＲＣ６と同等の乱数列の生成能力を有する。なお、本発明においては、上記のＲＣ６同等の写像の以外の写像を選択できるため、さまざまなバリエーションの乱数を得ることができる。

【００６３】

また、これ以外の演算や値を選択した場合にも、良好な乱数列が得られることが、実験により実証されている。

【００６４】

図１は、本実施形態に係る乱数生成装置の概要構成を示す模式図である。図２は、本実施形態に係る乱数生成装置において実行させる処理の制御の流れを示すフローチャートである。以下、これらの図を参照して、本実施形態について、詳細に説明する。

【００６５】

乱数列生成装置１０１は、wビットの乱数の列を生成し、種受付部１０２と、初期化部１０３と、変換部１０４と、回転部１０５と、更新部１０６と、出力部１０７と、を備える。

【００６６】

まず、乱数列生成装置１０１の種受付部１０２は、wビットの整数の列$s_1, s_2, \cdots, s_n, \cdots, s_m$を種として受け付ける（ステップＳ２０１）。ただし、$1 \leqq n \leqq m-1$である。

【００６７】

典型的には、$s_1, s_2, \cdots, s_n, \cdots, s_m$は、乱数列生成装置が有するＲＡＭ（Random Access Memory）等のメモリに格納されるが、ＣＰＵが有するキャッシュに格納してもよいし、ハードディスク等の読み書き可能な外部記録媒体に一時的に記憶してもよい。

【００６８】

ついで、初期化部１０３は、当該受け付けられた$s_1, s_2, \cdots, s_n, \cdots, s_m$を、整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として変換部１０４に与える（ステップＳ２０２）。

【００６９】

ここで、$x_1, x_2, \cdots, x_n, \cdots, x_m$も同様に、ＲＡＭ等のメモリに格納される。この場合、初期化部１０３が実行する処理は、$s_1, s_2, \cdots, s_n, \cdots, s_m$に対応するメモリから$x_1, x_2, \cdots, x_n, \cdots, x_m$に対応するメモリへの値の転送によって実現することができる。

【００７０】

さらに、変換部１０４は、当該与えられた整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$のそれぞれに対して上記の非線型変換$g(\cdot, \cdot)$により定義される変換を施してwビットの整数の列$y_1, y_2, \cdots, y_n, \cdots, y_m$を得る（ステップＳ２０３）。

【００７１】

当該変換としては以下のような漸化式により定義されるものを採用することができる。

【００７２】

（１）整数i（$1 \leqq i \leqq m-1$）についての漸化式

$y_1 = g(x_m, x_1)$；

$y_{i+1} = g(x_i, x_{i+1})$

により変換を行う。

【００７３】

（２）所定の整数cを用いて、整数i（$1 \leqq i \leqq m-1$）についての漸化式

$y_1 = g(c, x_1)$；

$y_{i+1} = g(y_i, x_{i+1})$

により変換を行う。

【００７４】

（３）整数i（$1 \leqq i \leqq m-1$）についての漸化式

$y_1 = g(c, x_1)$；

$y_{i+1} = g(x_i, x_{i+1})$

により変換を行う。

【００７５】

これらの計算は、ＣＰＵが有するＡＬＵ（Arithmetic Logic Unit）を用いて実現することができる。$y_1, y_2, \cdots, y_n, \cdots, y_m$もまた、メモリ等に格納される。

【００７６】

そして、回転部１０５は、当該整数の列$y_{n+1}, \cdots, y_m$から回転ビット数を得る（ステップＳ２０４）。回転ビット数を得る手法としては、以下のようなものが考えられる。

【００７７】

すなわち、y_{n+1}，…，y_mをビット列と見て、あらかじめ定めたビット位置のビットを順に並べ、再度これを整数値と見る。図３に、w=4，m-n=2の場合に、所定のビット位置の値から整数値を得る様子を示す。本図の例では、8ビットのビット列から3ビット分を抽出している。

【００７８】

3ビットから得られる整数値は、符号無し整数と見れば、0～7の８通りである。この場合は、回転方向として「あらかじめ定めた方向（右もしくは左）」を採用し、得られる整数値をそのまま回転ビット数とする。

【００７９】

一方、１ビットを符号（正または負に対応付ける）とし、他の２ビットから回転量を得ることとして、この場合は、正の場合は左方向へ、負の場合は右方向へ、それぞれ絶対値分のビット数だけ回転するとしても良い。

【００８０】

ついで、回転部１０５は、当該$y_1, y_2, \cdots, y_n, \cdots, y_m$をwmビットのビット列と見て、これ、もしくは、これの一部に対して所定の回転演算を行って、得られたwmビットのビット列からwビットの整数の列$z_1, z_2, \cdots, z_n, \cdots, z_m$を得る（ステップＳ２０５）。

【００８１】

所定の回転演算としては、以下のようなものを採用することができる。

【００８２】

（１）wnビットのビット列を、得られた回転ビット数だけ巡回シフトするもの。図４に、この巡回シフトとして、w=4，n=4であって、$y_1, y_2, \cdots, y_4$をビッグエンディアンに配列して、１ビット左にシフトする場合の概要構成を示した。これは、wmビットのビット列の一部を回転する回転演算である。

【００８３】

（２）wmビットのビット列全体を、得られた回転ビット数だけ巡回シフトするもの。図４におけるwnビットのビット列の回転と同様に、wmビットのビット列全体を巡回させれば良い。

【００８４】

（３）y_1，…，y_nのwnビットのビット列を、得られた回転ビット数だけ巡回シフトするほか、これとは別に、y_{n+1}，…，y_mのw(m-n)ビットのビット列を、得られた回転ビット数だけ巡回シフトするもの。

【００８５】

これらの手法は、メモリ等に格納された$y_1, y_2, \cdots, y_n, \cdots, y_m$の全体もしくは一部を、を、ＣＰＵにとって自然なビット幅単位で、桁上がり・桁下がりを考慮しつつ、順次巡回シフトすることによって実現することができる。この場合、得られる$z_1, z_2, \cdots, z_n, \cdots, z_m$は、$y_1, y_2, \cdots, y_n, \cdots, y_m$が格納されていたメモリ内の領域に新たな値として格納されることとなる。

【００８６】

さらに、出力部１０７は、変換部１０４における変換ならびに回転部１０５における回転が、所定の回数繰り返されたか否かを判定する（ステップＳ２０６）。

【００８７】

たとえば、ステップＳ２０１の前において、メモリ内に用意されたカウンタ変数に「所定の回数の値」を代入し、ステップＳ２０４とステップＳ２０６の間において当該カウンタ変数の値を１減じ、ステップＳ２０６においては、当該カウンタ変数の値が０になったか否かを判定することによって、実現することができる。

【００８８】

所定の回数繰り返された場合（ステップＳ２０６；Ｙｅｓ）最後に得られた$z_1, z_2, \cdots, z_n$を、乱数列$r_1, r_2, \cdots, r_n$として出力して（ステップＳ２０７）、乱数列の生成を終了する。

【００８９】

一方、所定の回数繰り返されていない場合（ステップＳ２０６；Ｎｏ）、更新部１０６は、当該$z_1, z_2, \cdots, z_n, \cdots, z_m$を整数列$x_1, x_2, \cdots, x_n, \cdots, x_m$として変換部１０４に与えて（ステップＳ２０８）、ステップＳ２０３に戻り、変換（ステップＳ２０３）、回転（ステップＳ２０４、Ｓ２０５）の処理を繰り返す。

【００９０】

これは、$z_1, z_2, \cdots, z_n, \cdots, z_m$が格納されているメモリ等内の値を、$x_1, x_2, \cdots, x_n, \cdots, x_m$が格納されているメモリ等へ転送することによって実現できる。

【００９１】

乱数列生成装置１０１においては、図示しない記憶部が存在し、その記憶部は、$s_1, s_2, \cdots, s_n, \cdots, s_m$、$x_1, x_2, \cdots, x_n, \cdots, x_m$、$y_1, y_2, \cdots, y_n, \cdots, y_m$、$z_1, z_2, \cdots, z_n, \cdots, z_m$、$r_1, r_2, \cdots, r_n, \cdots, r_m$などを異なる領域に、もしくは、値の利用の依存関係を分析することにより、同じ領域に記憶する（たとえば、$y_1, y_2, \cdots, y_n, \cdots, y_m$と$z_1, z_2, \cdots, z_n, \cdots, z_m$等。）ような構成をとることができる。また、各部は、上記の共有メモリを用いて互いに計算した値をやりとりするのである。

【００９２】

図５は、本実施形態の乱数列生成装置１０１が実現されるコンピュータの典型的な概要構成を示す模式図である。以下、本図を参照して説明する。

【００９３】

コンピュータ３０１は、ＣＰＵ　３０２によって制御される。コンピュータ３０１に電源が投入されると、ＣＰＵ　３０２は、ＲＯＭ（Read Only Memory）３０３に用意されたＩＰＬ（Initial Program Loader）を実行する。

【００９４】

そして、ＩＰＬの実行により、フレキシブルディスクドライブ３０４に装着されたフレキシブルディスクやハードディスク３０５等に記録されたＯＳ（Operating System）がロードされ、ユーザからの各種の指示入力を受け付けることができるようになる。

【００９５】

ユーザは、キーボード３０６やマウス３０７を操作して、コンピュータ３０１に対して各種の指示入力を与える。

【００９６】

これに応じて、ＯＳは、ハードディスク３０５やＣＤ－ＲＯＭ（Compact Disk ROM）ドライブ３０８に装着されたＣＤ－ＲＯＭに記録されたプログラムや各種のデータをＣＰＵ　３０２に実行させ、実行の過程やその結果をディスプレイ３

０９に表示する。

【００９７】

また、ＣＰＵ　３０２は、一時的な記憶域として、ＲＡＭ　３１１を利用する。ＲＡＭ　３１１は、上記のように、計算の途中で利用される各種の数列を記憶するために用いられる。

【００９８】

さらに、ＣＰＵ　３０２は、プログラムの実行の過程において、ハードディスク３０５に、生成された乱数列などの処理の結果や途中経過などの情報を保存することができる。

【００９９】

なお、本実施形態における演算は、上記のように、きわめて単純なビット演算に還元することができる。したがって、専用の電子回路（加算器、減算器、シフタ、ラッチ等）を組み合わせて乱数列生成装置１０１を構成することができるほか、ＡＳＩＣ（Application Specific Integrated Circuit）、ＤＳＰ（Digital Signal Processor）やＦＰＧＡ（Field Programmable Gate Array）などのような電子回路の構成状態を可変に変更できる電子素子を利用して、乱数列生成装置１０１を構成することができ、これらの態様も本発明の範囲に含まれる。

【０１００】

（実験の結果）

上記実施形態に係る乱数列生成装置１０１を用いて、以下の諸元で乱数列を生成させた。

$W = 32,$

$n = 32,$

$g(a,b) = 2b^2 + h(a)b$

【０１０１】

ただし、写像h(・)は、与えられた値の数値表現の最下位２ビットを０１に置換する演算により定義される。

【０１０２】

また、変換および回転は、それぞれ１ラウンドごとに１回とした。すなわち、

「所定の回数の繰り返し」は１回である。

【０１０３】

出力されるのは、全部でwn = 1024ビットの乱数列$r_1, r_2, \cdots, r_{1024}$である。

【０１０４】

これに対して２００００×８９９９９種類の種を与え、２００００×８９９９９ラウンドだけ乱数列$r_1, r_2, \cdots, r_{1024}$を出力させた。

【０１０５】

そして、これに対して乱数列のランダム性を検査する標準的なテストであるＦＩＰＳ　１４０－１ならびにＦＩＰＳ　１４０－２のうち、標準セキュリティ規格にあるランダム性検査テストを、１０２４ビットの乱数列の中の各ビット位置毎に適用して、本実施形態の乱数列の性質を検査した。

【０１０６】

これらのテストにおいては、各ビット位置から２００００ビットのビット列を取り出し、その２００００ビットのビット列に対して、以下が行われる。

モノビットテスト（monobit test）。所定の位置のビットの値の出現頻度に偏りがないか否かを調べるもの。

ポーカーテスト（poker test）。２００００ビットを５０００個の４ビットパターンに分割し、その４ビットパターンの出現頻度に偏りがないか否かを調べるもの。

ランズテスト（runs test）。乱数列から所定の長さの連を切り出した場合に、当該長さの連の出現頻度に偏りがないか否かを調べるもの。長さとしては１～６を用いる。

ロングランズテスト（long runs teset）。ランズテストと同様であるが、ＦＩＰＳ　１４０－１の場合は３４以上の連が存在する場合ランダム性が否定され、ＦＩＰＳ　１４０－１の場合は２６以上の連が存在する場合ランダム性が否定される。

【０１０７】

実験の結果、ＦＩＰＳ　１４０－１においては、生成された１０２４ビット×８９９９９サンプルの２００００ビット列は、すべて、定められた基準をクリア

した。

【０１０８】

また、ＦＩＰＳ　１４０－２においては、生成された１０２４ビット×８９９９サンプルの２００００ビット列の内、９９．９２パーセントのサンプルが、定められた基準をクリアした。

【０１０９】

さらに、上記の乱数テストよりも厳しい乱数テストであるＮＩＳＴ　８００－２２に本発明を適用して調べた結果、回転の手法（３）を利用すると、極めて良好な乱数を得られることが判明している。

【０１１０】

また、本技術をＸＩＬＩＮＸ（登録商標）社のVertex xcv1000（システムゲート数は１００万）のＦＰＧＡに実装したところ、本アルゴリズムの並列性から、25.62Gビット/secのスピードで乱数列を生成させることができた。すなわち、本技術をＦＰＧＡ等のハードウェアに実装すると、高速性の上で多大なメリットを得ることができる。

【０１１１】

このようにして、本実施形態によって生成された乱数列は極めて性質が良いものであり、秘密通信の暗号化の分野や物理現象、化学現象などの模擬実験の分野で有用であり、ハードウェア上で高速に良好なランダム性を持つ乱数列を出力するのに極めて有用であることが示された。

【０１１２】

（暗号化復号化装置）

上記の乱数生成装置を利用すれば、暗号化や復号化を実現することができる。図６は、このような暗号化や復号化を行う暗号化装置および復号化装置の概要構成を示す模式図である。

【０１１３】

暗号化装置６０１と、復号化装置６５１と、は、$s_1, \cdots, s_n, \cdots, s_m$を共通鍵とする。そして、それぞれが備える生成部６０２、６５２は、いずれも、同じ構成（計算手法）による乱数発生装置２０１を有し、この共通鍵$s_1, \cdots, s_n, \cdots, s_m$を入力

として受け付ける。すると、同じ乱数$r_1, \cdots, r_n$が出力されることとなる。

【０１１４】

この乱数を用いて、暗号化装置６０１においては、メッセージ受付部６０３が受け付けた伝送メッセージの整数列$p_1, p_2, \cdots, p_i, \cdots$を、ＸＯＲ部６０４が、$p_1$ xor r_1, p_2 xor $r_2, \cdots, p_i$ xor $r((i+n-1) \bmod n) + 1, \cdots$のように変換して、これを暗号化済メッセージの整数列$e_1, e_2, \cdots, e_i, \cdots$として出力する。

【０１１５】

ここで、xorは上述したように排他的論理和演算を意味し、a mod nはaをnで割った余りを意味する。

【０１１６】

一方、復号化装置６５１のメッセージ受付部６５３は、暗号化済メッセージの整数列$e_1, e_2, \cdots, e_i, \cdots$を受け付け、ＸＯＲ部６５４が$e_1$ xor r_1, e_2 xor $r_2, \cdots, e_i$ xor $r((i+n-1) \bmod n) + 1, \cdots$のように変換して、これを復号化済メッセージの整数列$f_1, f_2, \cdots, f_i, \cdots$として出力する。

【０１１７】

ここで、

$$f_i = e_i \text{ xor } r((i+n-1) \bmod n) + 1$$
$$= (p_i \text{ xor } r((i+n-1) \bmod n) + 1) \text{ xor } r((i+n-1) \bmod n) + 1$$
$$= p_i$$

であるから、復号化済メッセージの整数列ともとの伝送メッセージの整数列とは同じものであり、暗号化ならびに復号化ができることが示された。

【０１１８】

なお、処理の対象となるメッセージの長さは、n以下とすることもできる。その場合には、上記の$((i+n-1) \bmod n) + 1$はiに置き換えることができる。このようにすれば、同じ乱数列を繰り返し利用するよりも、さらに秘匿性を高めることができる。

【０１１９】

上記のように、暗号化装置６０１と復号化装置６５１とは全く同じ構成であるので、１つの装置をある場合は暗号化装置６０１として、ある場合は復号化装置

６５１として、それぞれ利用することができる。

【０１２０】

【発明の効果】

以上説明したように、本発明によれば、生成される乱数列が乱数列として好ましい性質を有するような乱数列生成装置、乱数列生成方法、これらを用いた暗号化復号化装置、暗号化復号化方法、ならびに、これらをコンピュータによって実現するためのプログラムを提供することができる。

【図面の簡単な説明】

【図１】

本発明の実施形態に係る乱数列生成装置の概要構成を示す模式図である。

【図２】

本実施形態の乱数列生成装置において実行される乱数列生成処理の制御の流れを示すフローチャートである。

【図３】

本実施形態の乱数列生成装置の回転部において実行される回転ビット数の取得の様子を示す説明図である。

【図４】

本実施形態の乱数列生成装置の回転部において実行される回転演算の様子を示す説明図である。

【図５】

本実施形態の乱数列生成装置が実現される典型的なコンピュータの概要構成を示す模式図である。

【図６】

暗号化装置と復号化装置の実施形態の概要構成を示す模式図である。

【符号の説明】

１０１　乱数列生成装置

１０２　種受付部

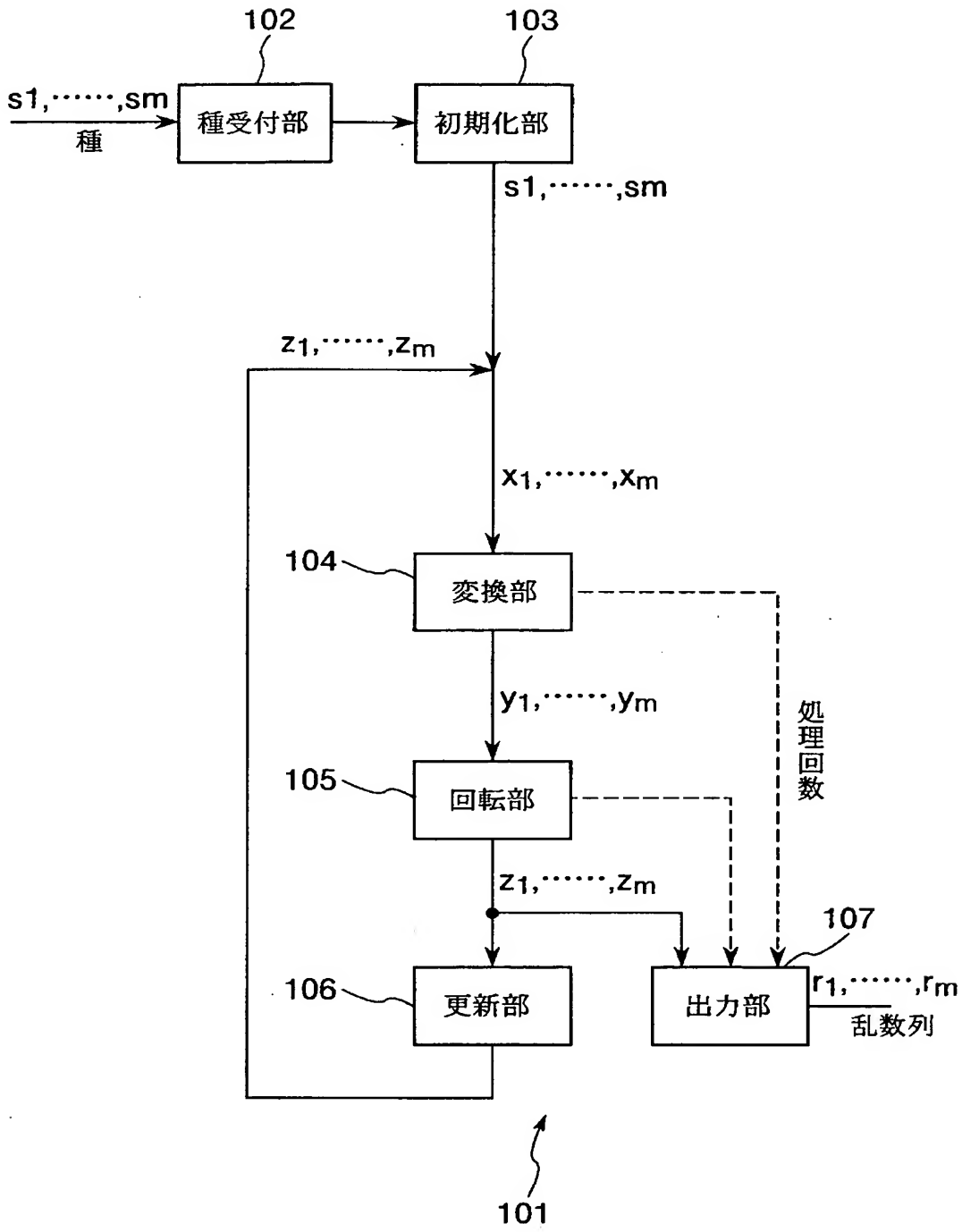
１０３　初期化部

１０４　変換部

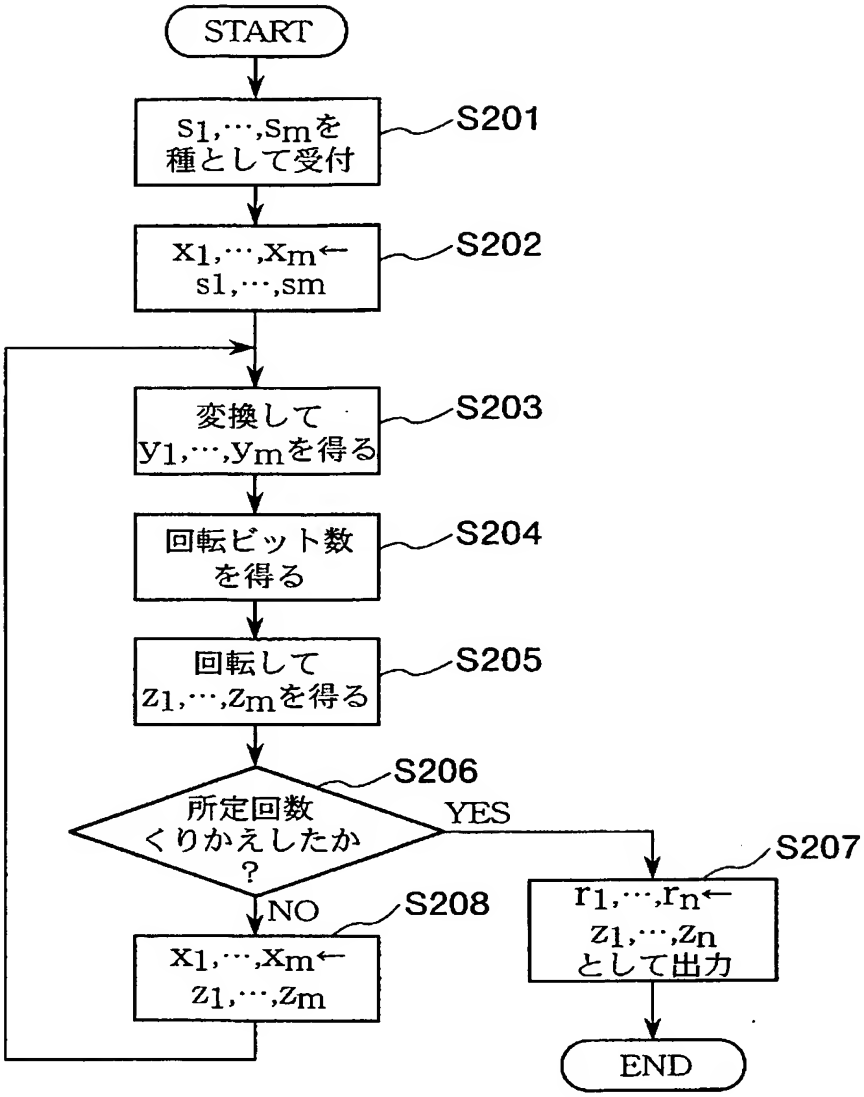
１０５　回転部
１０６　更新部
１０７　出力部
３０１　コンピュータ
３０２　ＣＰＵ
３０３　ＲＯＭ
３０４　フレキシブルディスクドライブ
３０５　ハードディスク
３０６　キーボード
３０７　マウス
３０８　ＣＤ－ＲＯＭドライブ
３０９　ディスプレイ
３１１　ＲＡＭ
６０１　暗号化装置
６０２　乱数生成部
６０３　メッセージ受付部
６０４　ＸＯＲ部
６５１　復号化装置
６５２　乱数生成部
６５３　メッセージ受付部
６５４　ＸＯＲ部

【書類名】図面

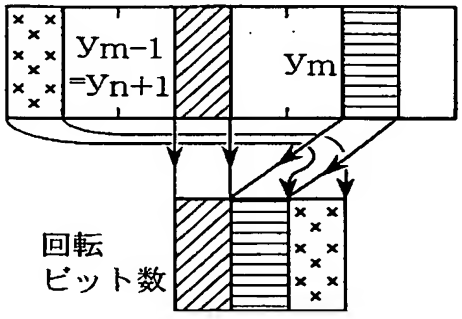
【図1】


102
103
s1,……,sm
種
種受付部
初期化部
s1,……,sm
z1,……,zm
x1,……,xm
104
変換部
y1,……,ym
処理回数
105
回転部
z1,……,zm
107
106
更新部
出力部
r1,……,rm
乱数列
101

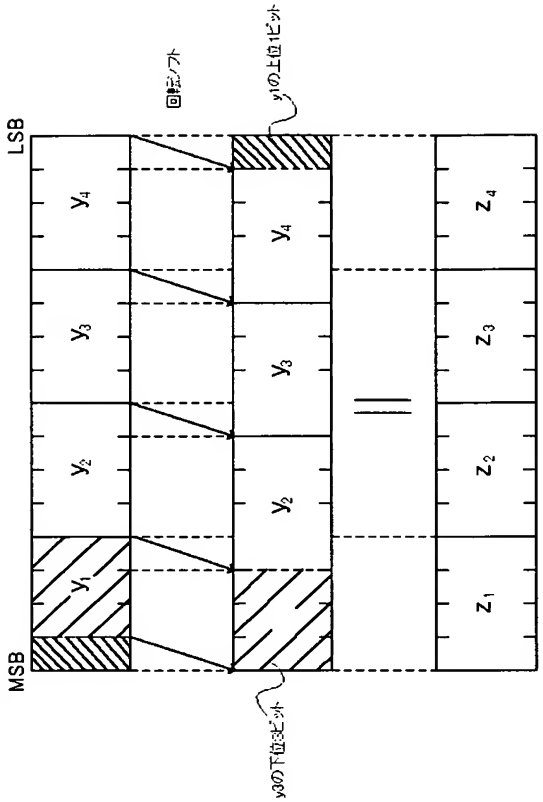
【図2】


START
s1,…,smを種として受付
S201
x1,…,xm← s1,…,sm
S202
変換して y1,…,ymを得る
S203
回転ビット数を得る
S204
回転して z1,…,zmを得る
S205
所定回数くりかえしたか？
S206
YES
NO
S207
r1,…,rn← z1,…,zn として出力
S208
x1,…,xm← z1,…,zm
END

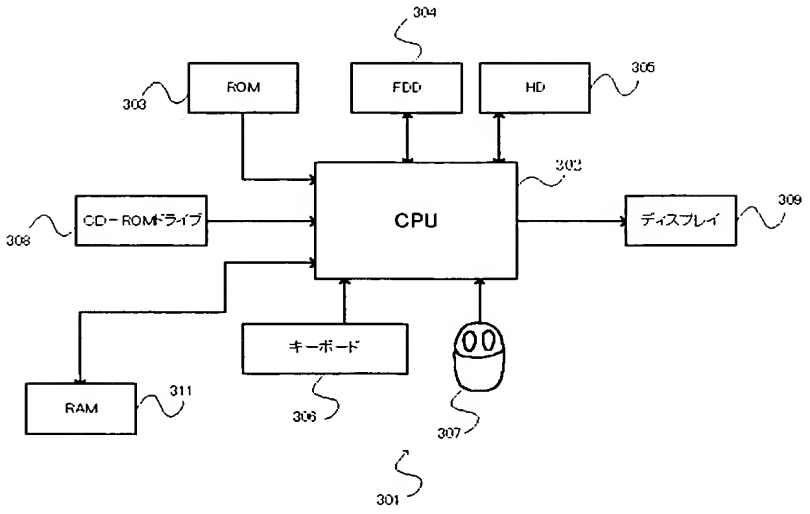
【図3】


ym−1 =yn+1
ym
回転ビット数

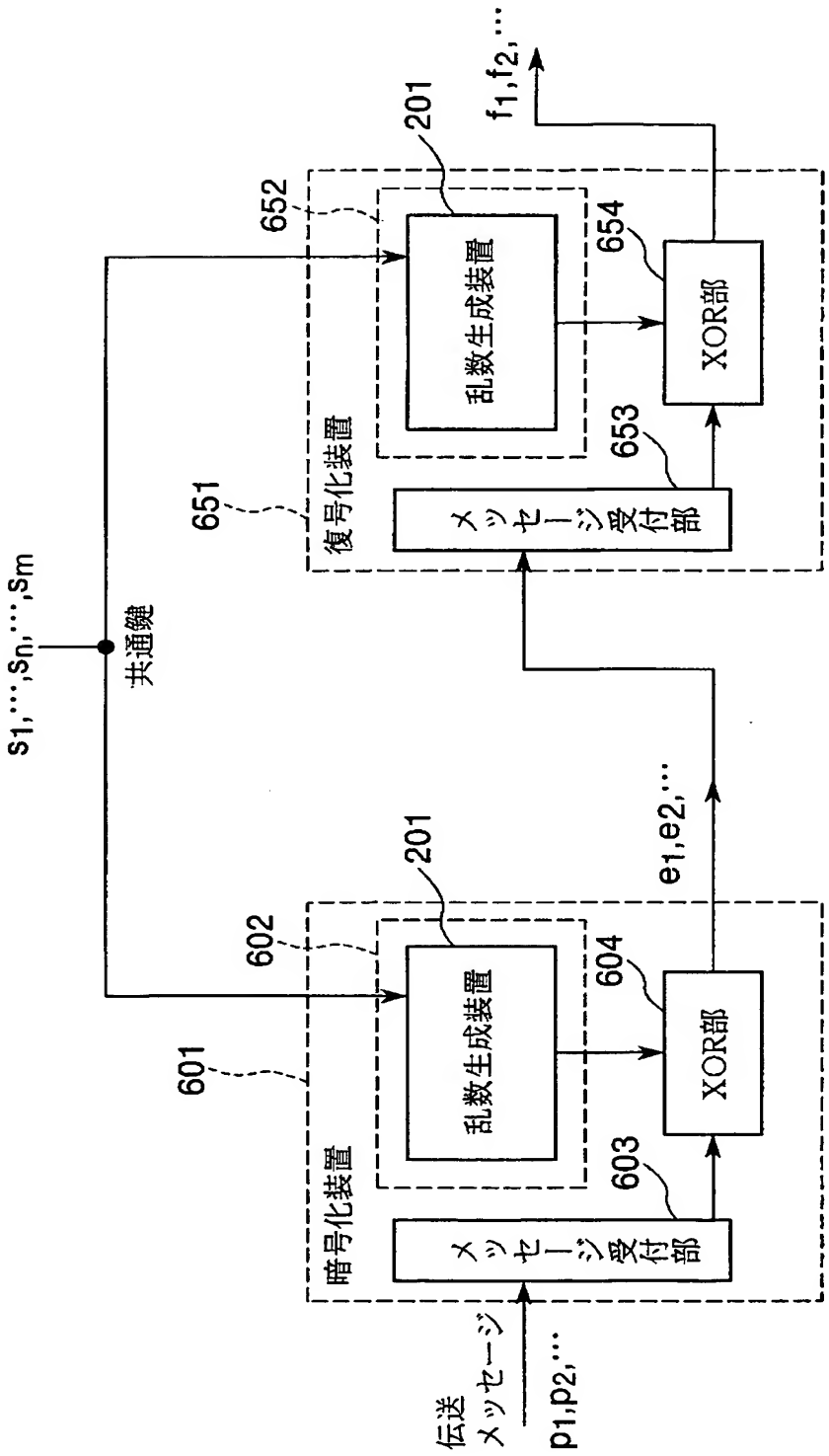
【図4】


LSB
MSB
y4
y3
y2
y1
z4
z3
z2
z1

【図5】


304
ROM
FDD
HD
305
303
302
CD－ROMドライブ
CPU
ディスプレイ
309
308
キーボード
RAM
311
306
307
301

【図6】


f1,f2,…
201
652
654
乱数生成装置
XOR部
653
651
復号化装置
メッセージ受付部
s1,…,sn,…,sm
共通鍵
e1,e2,…
201
602
604
乱数生成装置
XOR部
601
603
暗号化装置
メッセージ受付部
伝送メッセージ
p1,p2,…

【書類名】　要約書

【要約】

【課題】　乱数列生成装置等を提供する。

【解決手段】　乱数列生成装置１０１の種受付部１０２は、wビットの整数の列を種として受け付け、初期化部１０３は、当該受け付けられた整数の列を、変換部１０４に与え、変換部１０４は、当該与えられた整数列のそれぞれに対して所定の変換を施してwビットの整数の列を得て、回転部１０５は、得られた整数の列の一部から回転ビット数を得て、当該得られた整数の列をビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、wビットの整数の列を得て、更新部は、当該整数の列を変換部１０４に与え、出力部１０７は、変換部における変換ならびに回転部における回転が、所定の回数繰り返された場合、最後に得られた整数列の一部を乱数列として出力する。

【選択図】　図１

特願２００３－０７５４３８

出　願　人　履　歴　情　報

識別番号　［３０１０２２４７１］

１．変更年月日　２００１年　４月　２日
［変更理由］　新規登録
住　所　東京都小金井市貫井北町４－２－１
氏　名　独立行政法人通信総合研究所